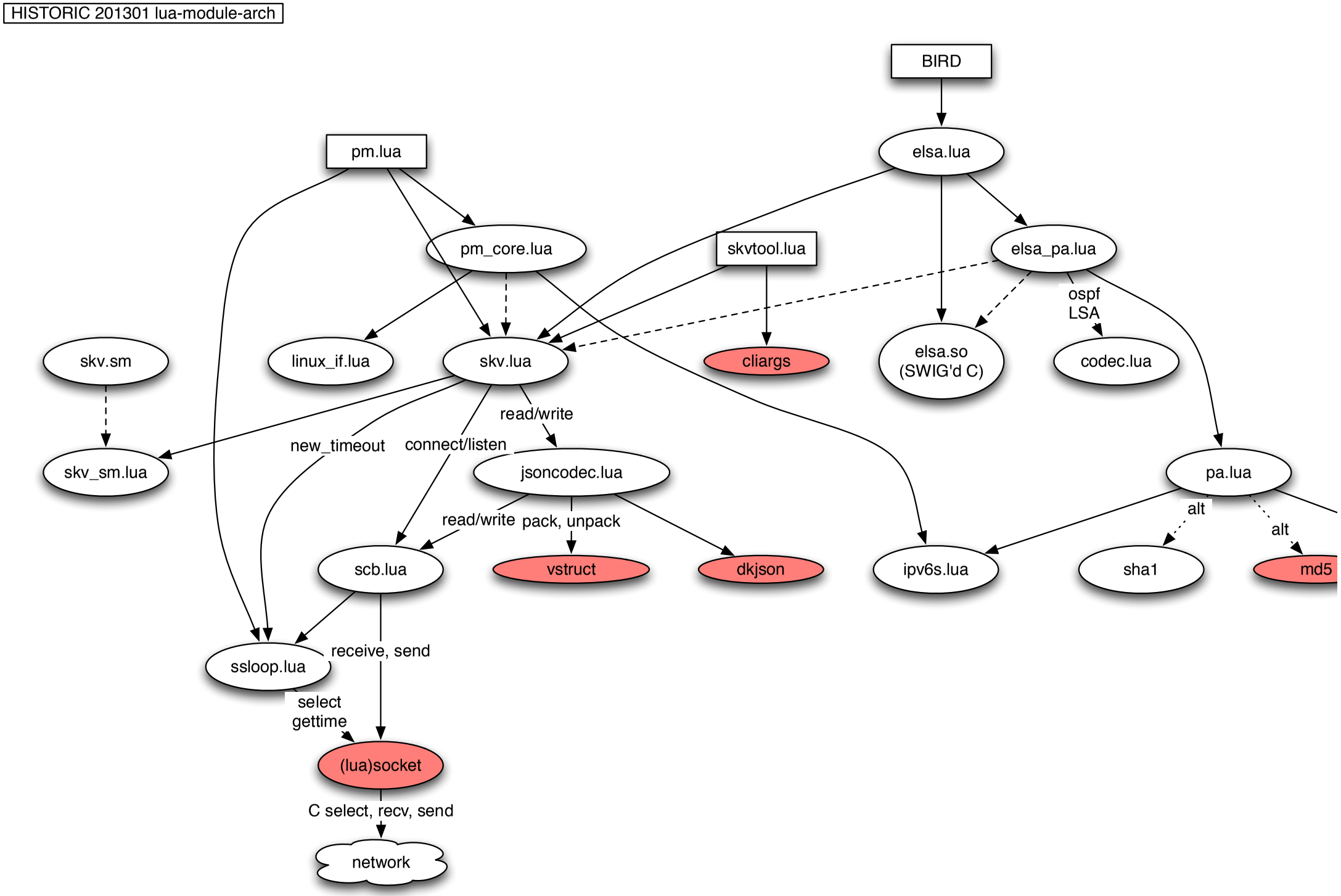
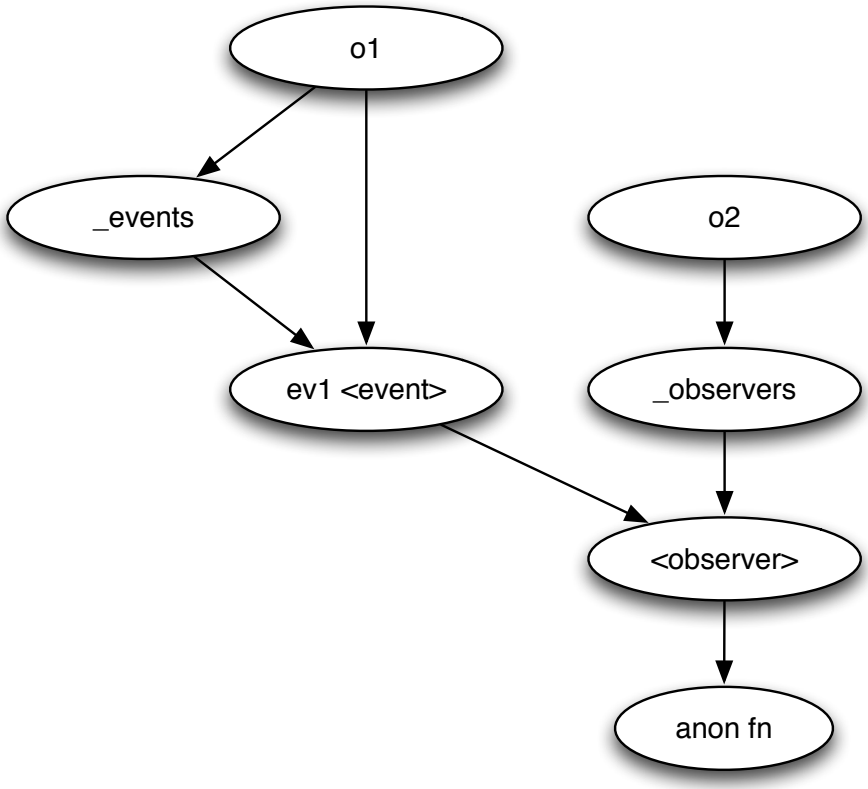
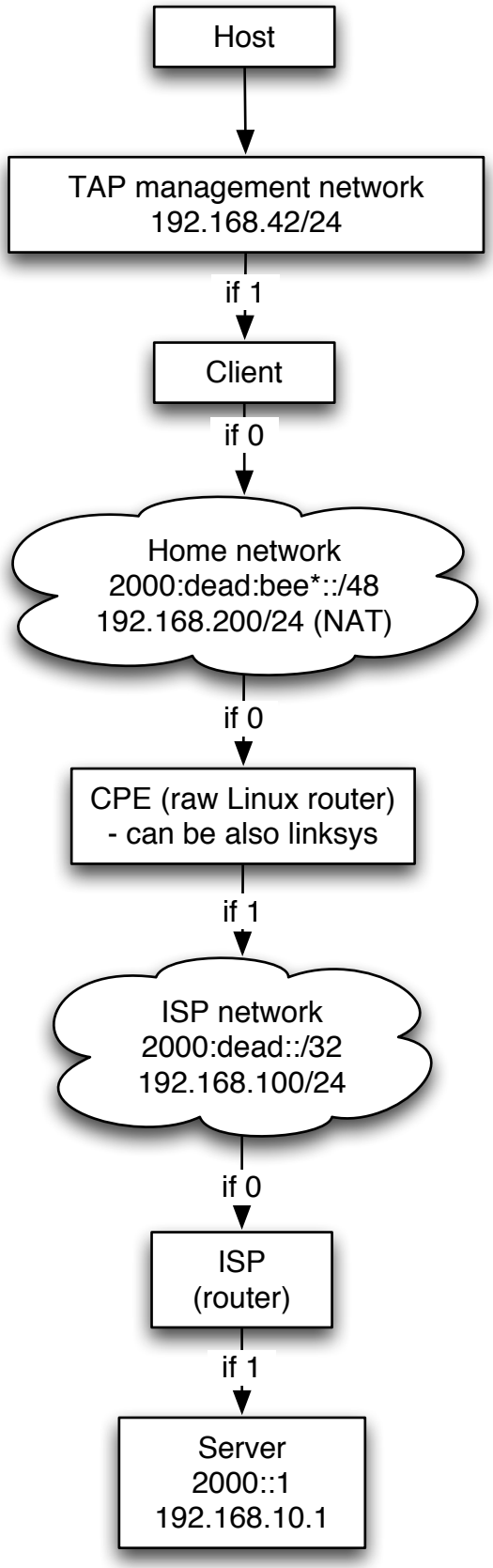

HISTORIC 201301 lua-module-arch
BIRD
pm.lua
elsa.lua
pm_core.lua
skvtool.lua
elsa_pa.lua
ospf
LSA
skv.sm
linux_if.lua
skv.lua
cliargs
elsa.so
(SWIG'd C)
codec.lua
read/write
new_timeout
connect/listen
skv_sm.lua
jsoncodec.lua
pa.lua
alt
alt
read/write
pack, unpack
scb.lua
vstruct
dkjson
ipv6s.lua
sha1
md5
receive, send
ssloop.lua
select
gettime
(lua)socket
C select, recv, send
network

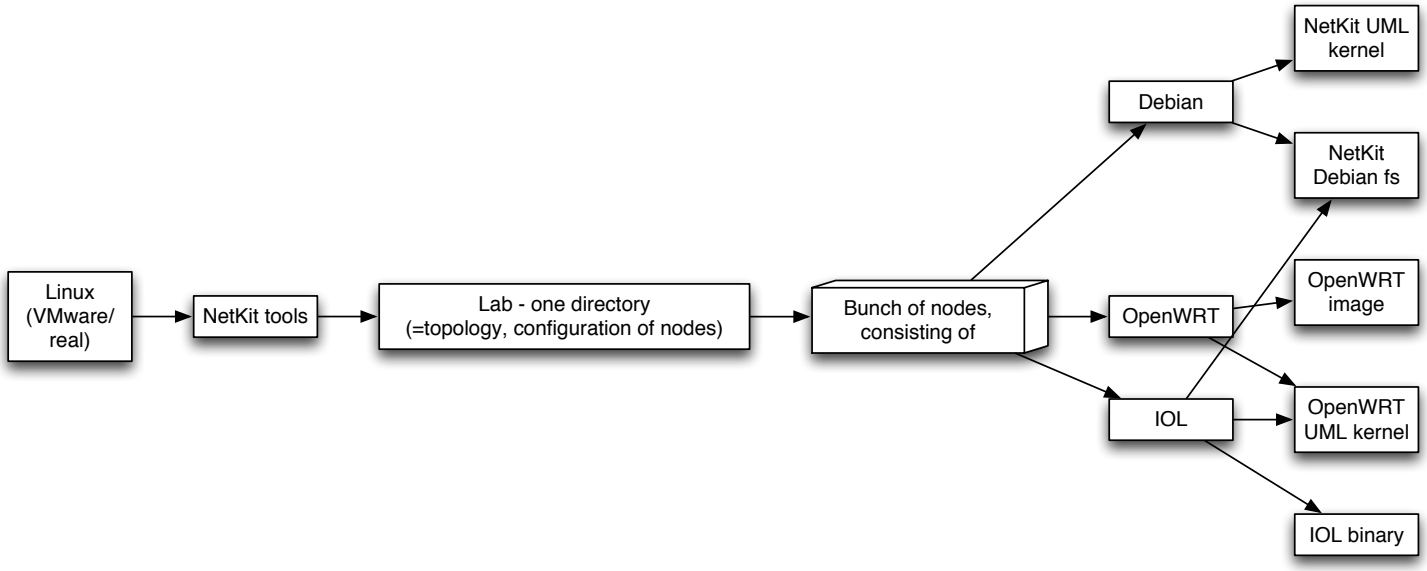
201209 lua-observer-pattern

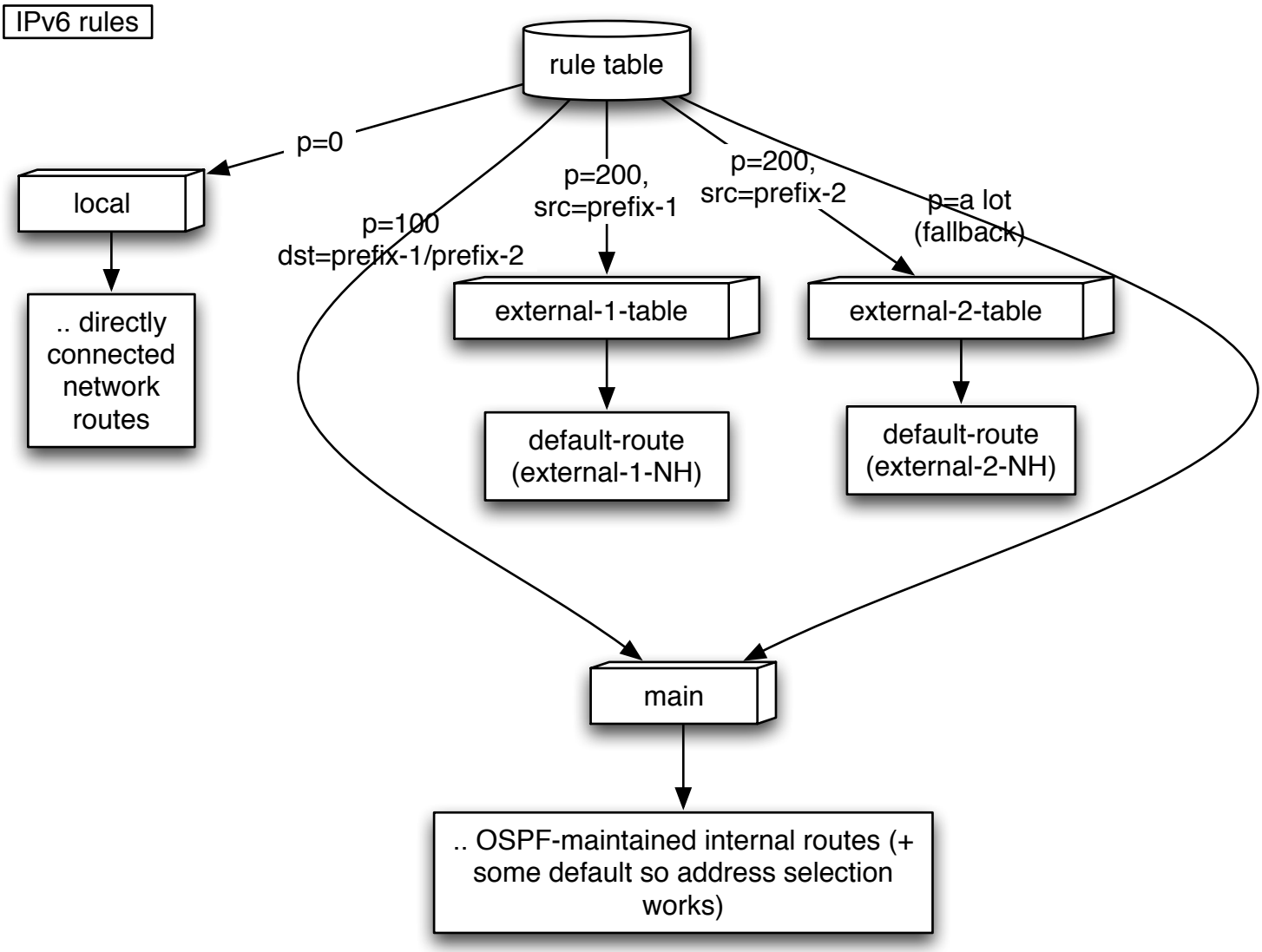


simple4+ setup

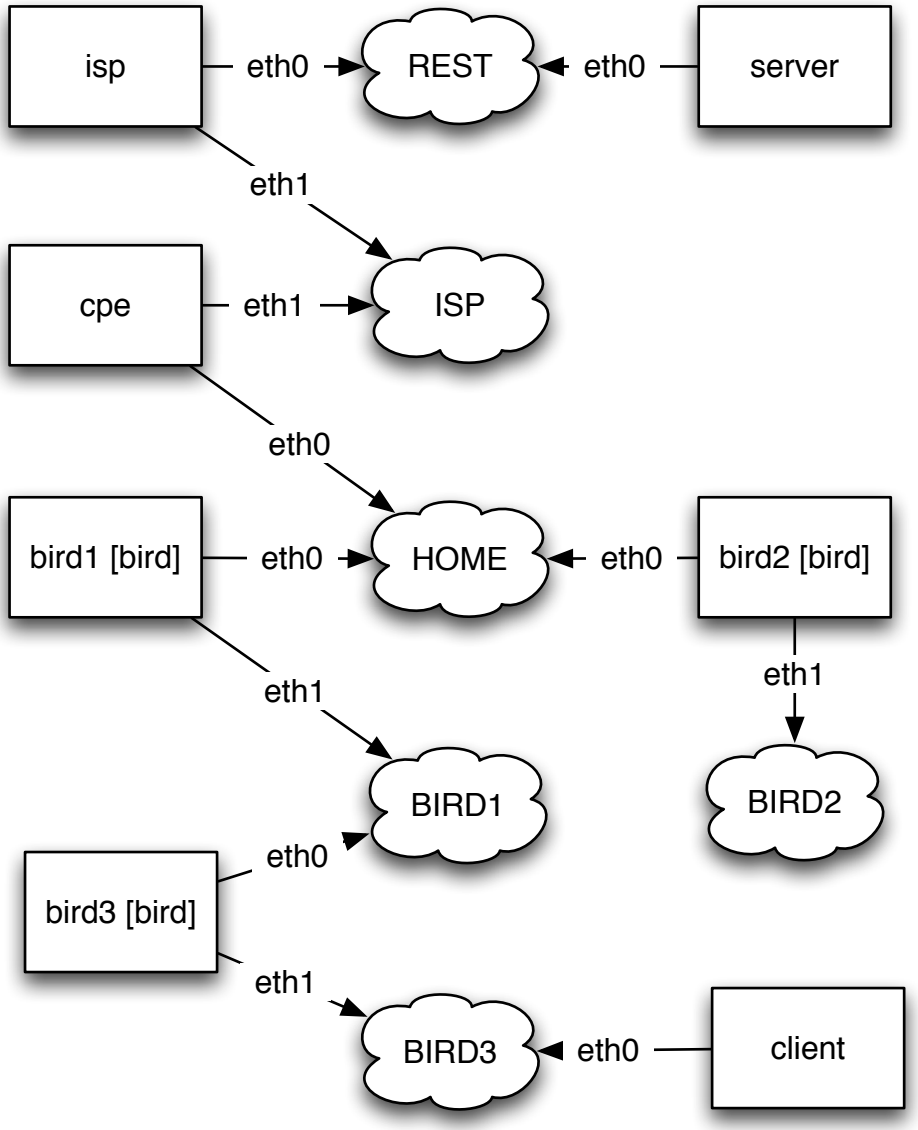


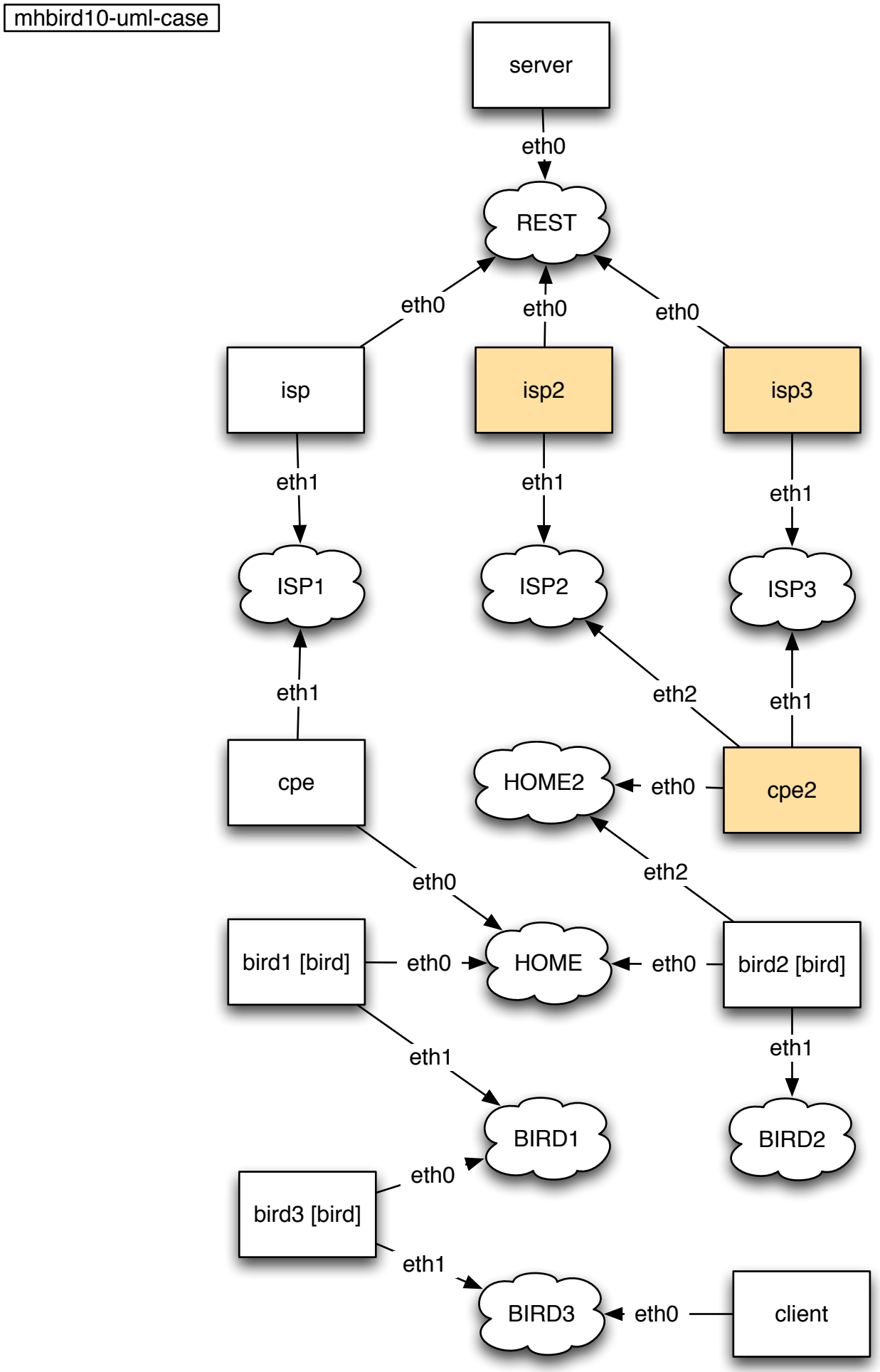
Dev infrastructure

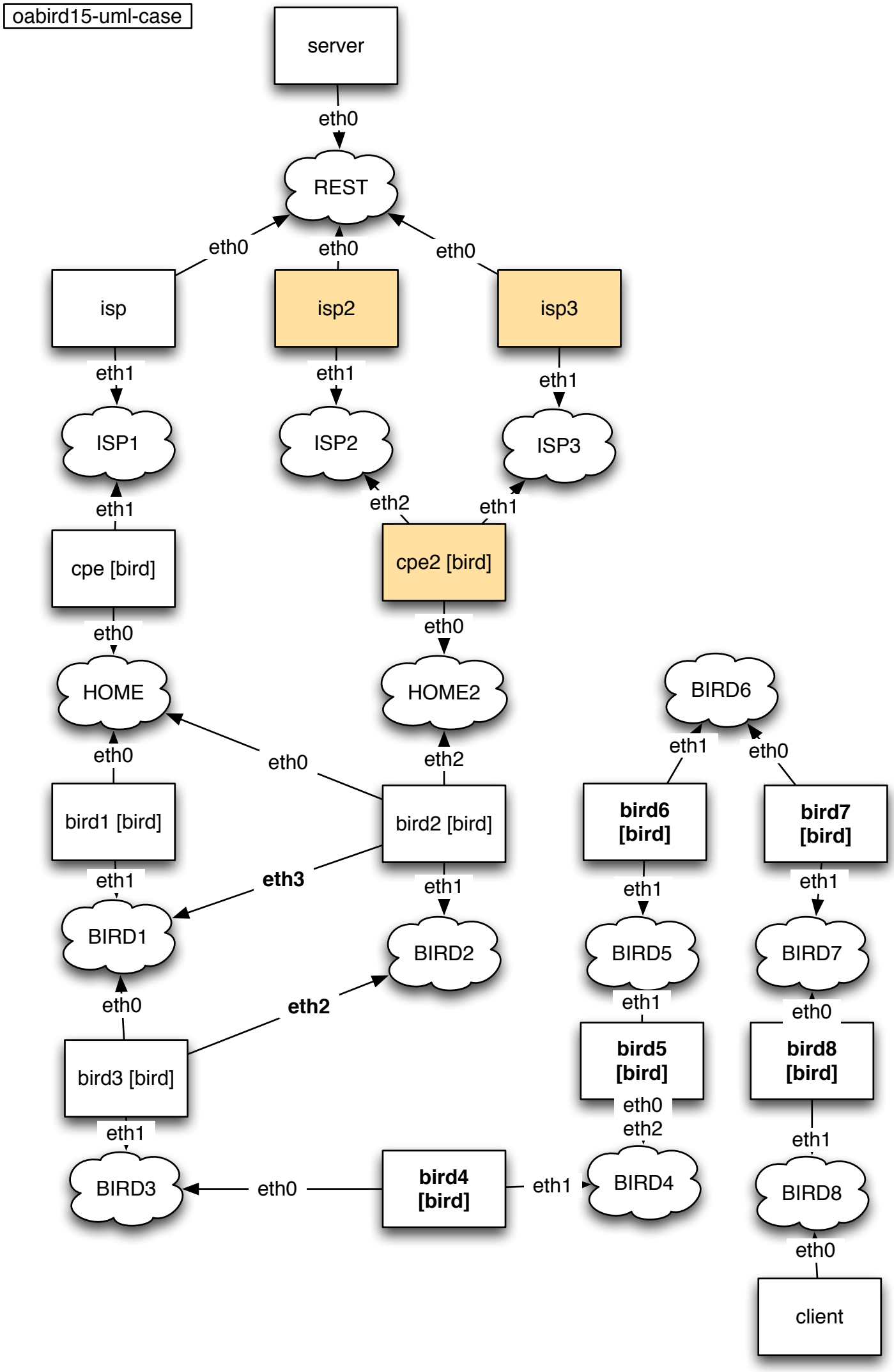



IPv6 rules
rule table
p=0
local
.. directly connected network routes
p=100
dst=prefix-1/prefix-2
p=200,
src=prefix-1
external-1-table
default-route (external-1-NH)
p=200,
src=prefix-2
external-2-table
default-route (external-2-NH)
p=a lot
(fallback)
main
.. OSPF-maintained internal routes (+ some default so address selection works)

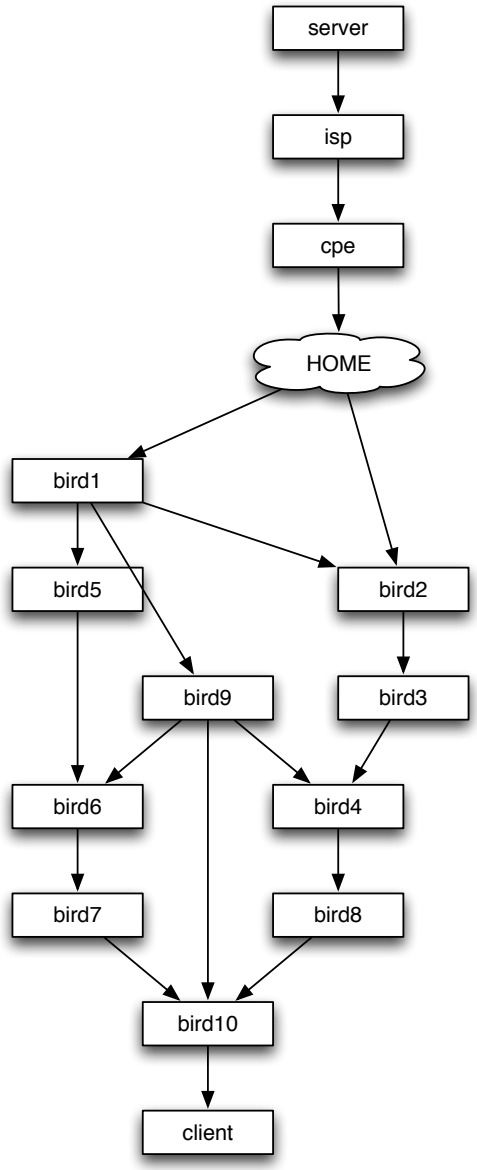
bird7-uml-case



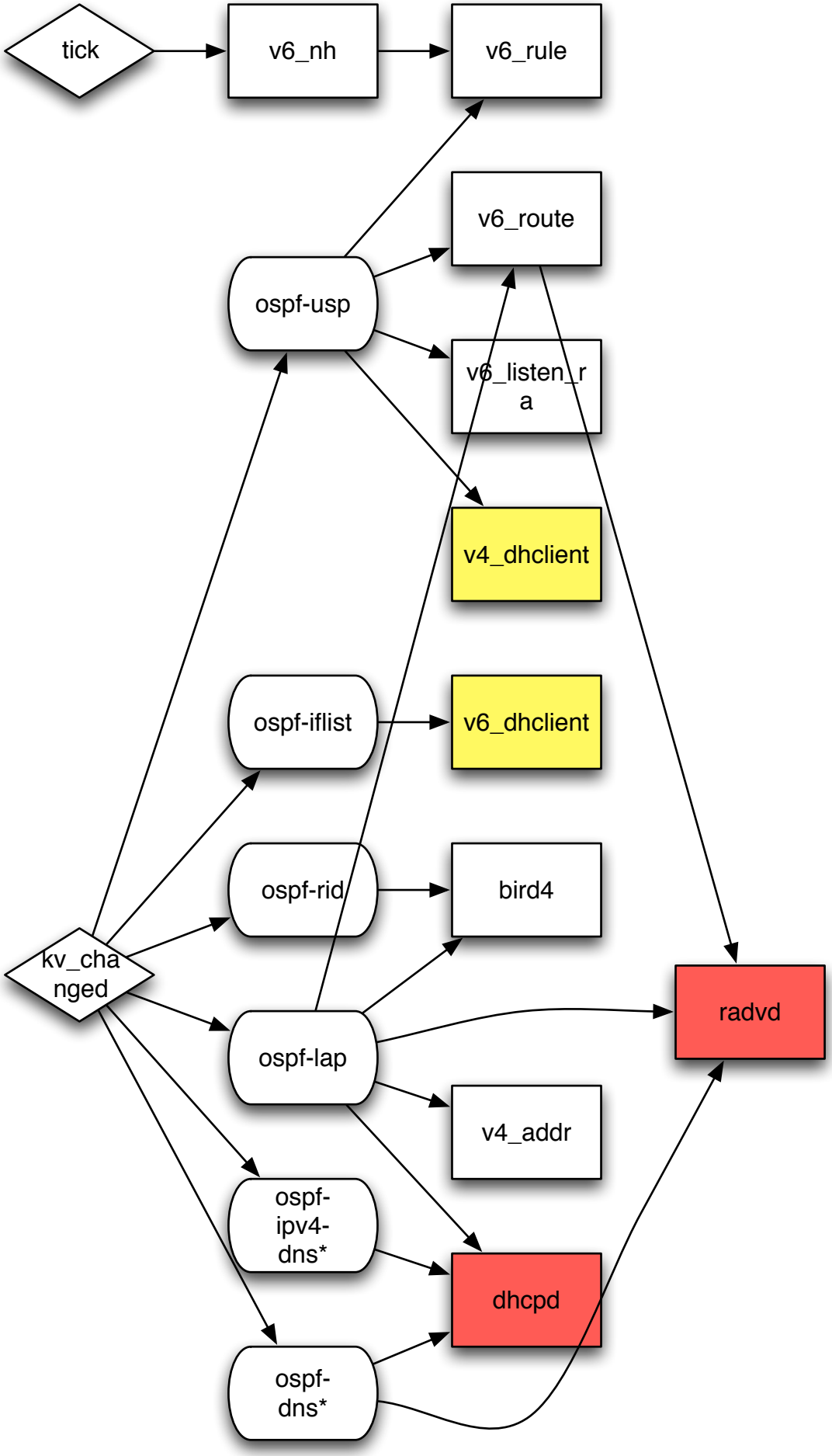

mhbird10-uml-case
server
eth0
REST
eth0
eth0
eth0
isp
isp2
isp3
eth1
eth1
eth1
ISP1
ISP2
ISP3
eth1
eth2
eth1
cpe
HOME2
eth0
cpe2
eth2
eth0
bird1 [bird]
eth0
HOME
eth0
bird2 [bird]
eth1
eth1
BIRD1
BIRD2
eth0
bird3 [bird]
eth1
BIRD3
eth0
client


oabird15-uml-case
server
eth0
REST
eth0
eth0
eth0
isp
isp2
isp3
eth1
eth1
eth1
ISP1
ISP2
ISP3
eth2
eth1
eth1
cpe [bird]
cpe2 [bird]
eth0
eth0
HOME
HOME2
BIRD6
eth1
eth0
eth0
eth2
eth0
bird1 [bird]
bird2 [bird]
bird6 [bird]
bird7 [bird]
eth1
eth3
eth1
eth1
eth1
BIRD1
BIRD2
BIRD5
BIRD7
eth0
eth2
eth1
eth0
bird3 [bird]
bird5 [bird]
bird8 [bird]
eth0
eth2
eth1
eth1
BIRD3
eth0
bird4 [bird]
eth1
BIRD4
BIRD8
eth0
client

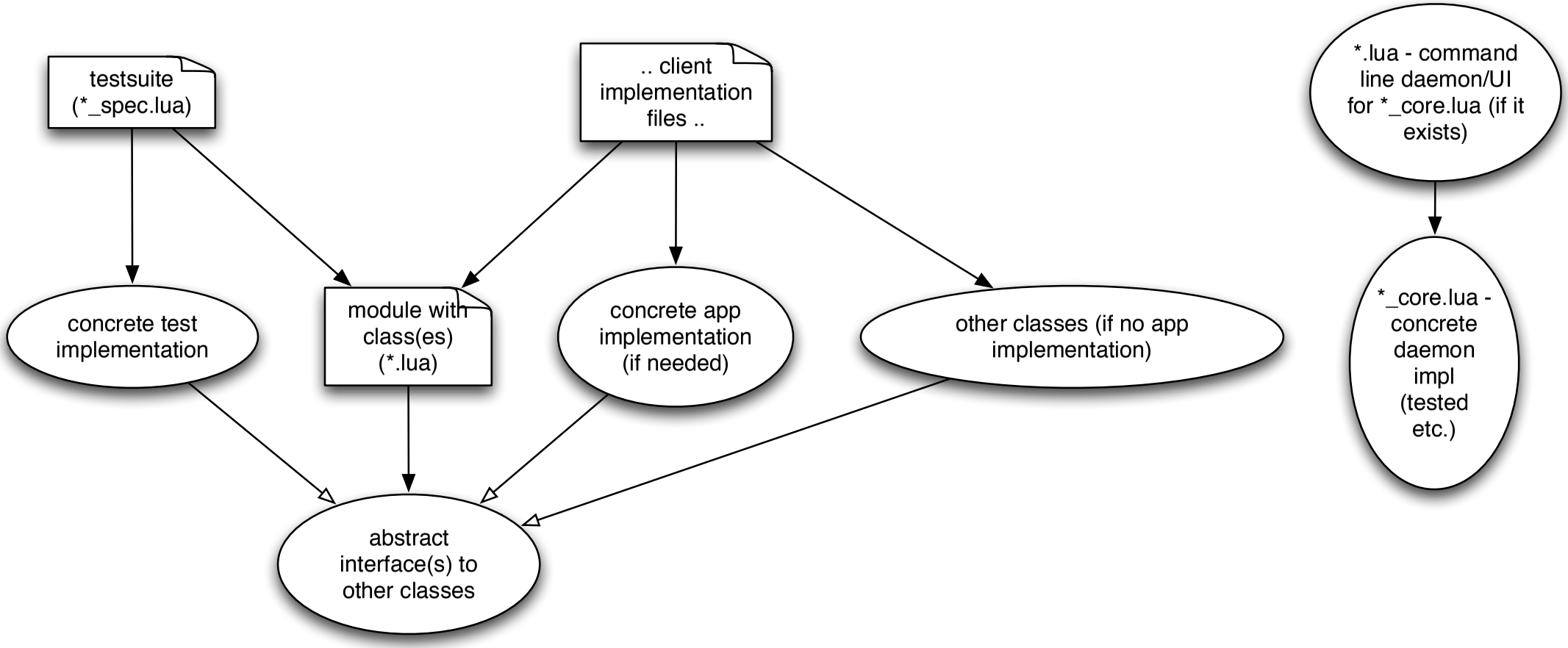
bird14


server
isp
cpe
HOME
bird1
bird5
bird2
bird9
bird3
bird6
bird4
bird7
bird8
bird10
client

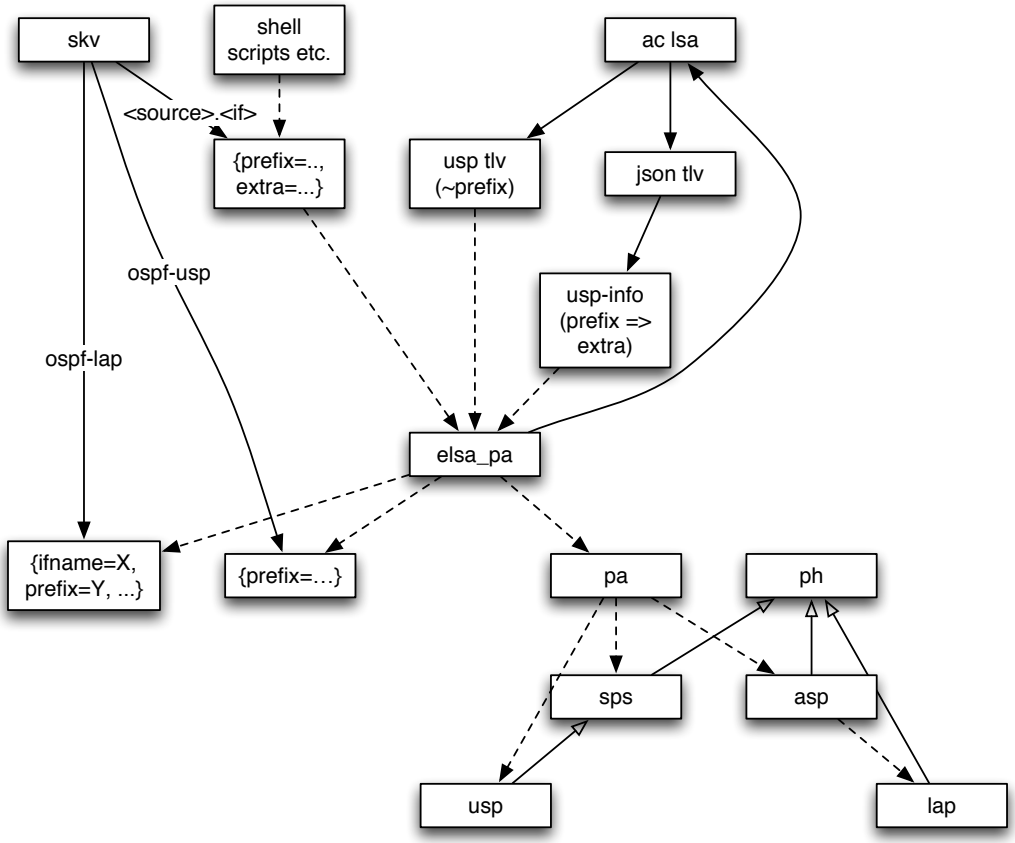
pm handler + key deps



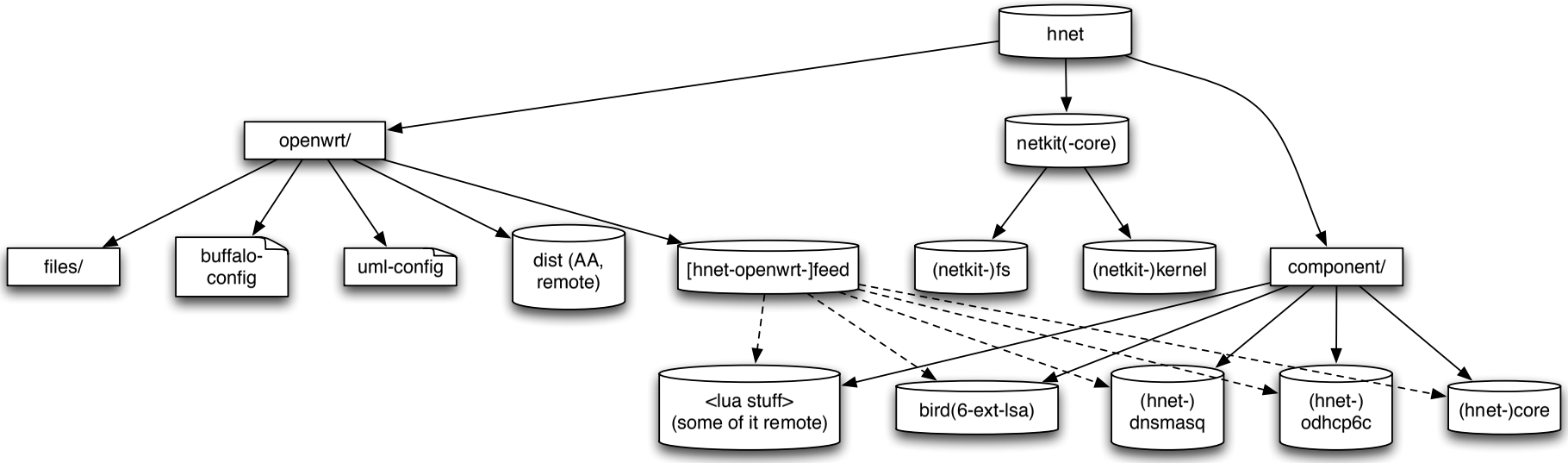
lua-general-arch



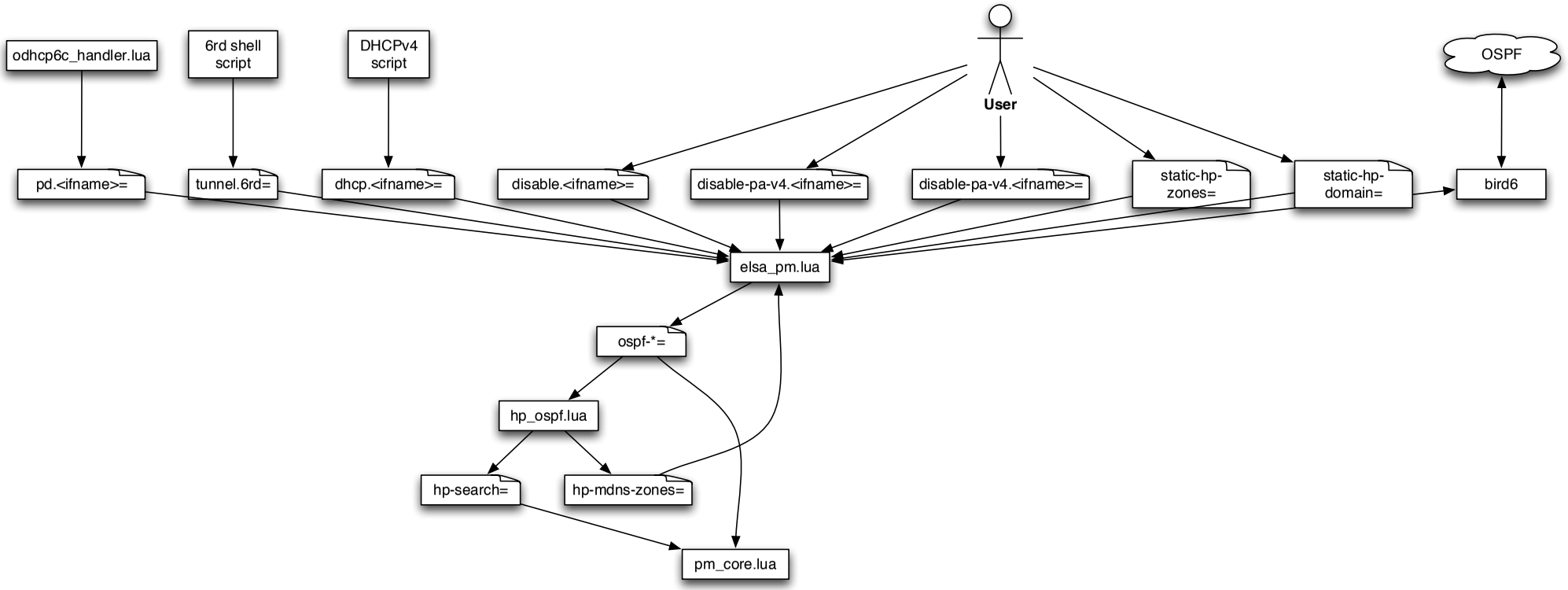
201302 data structures



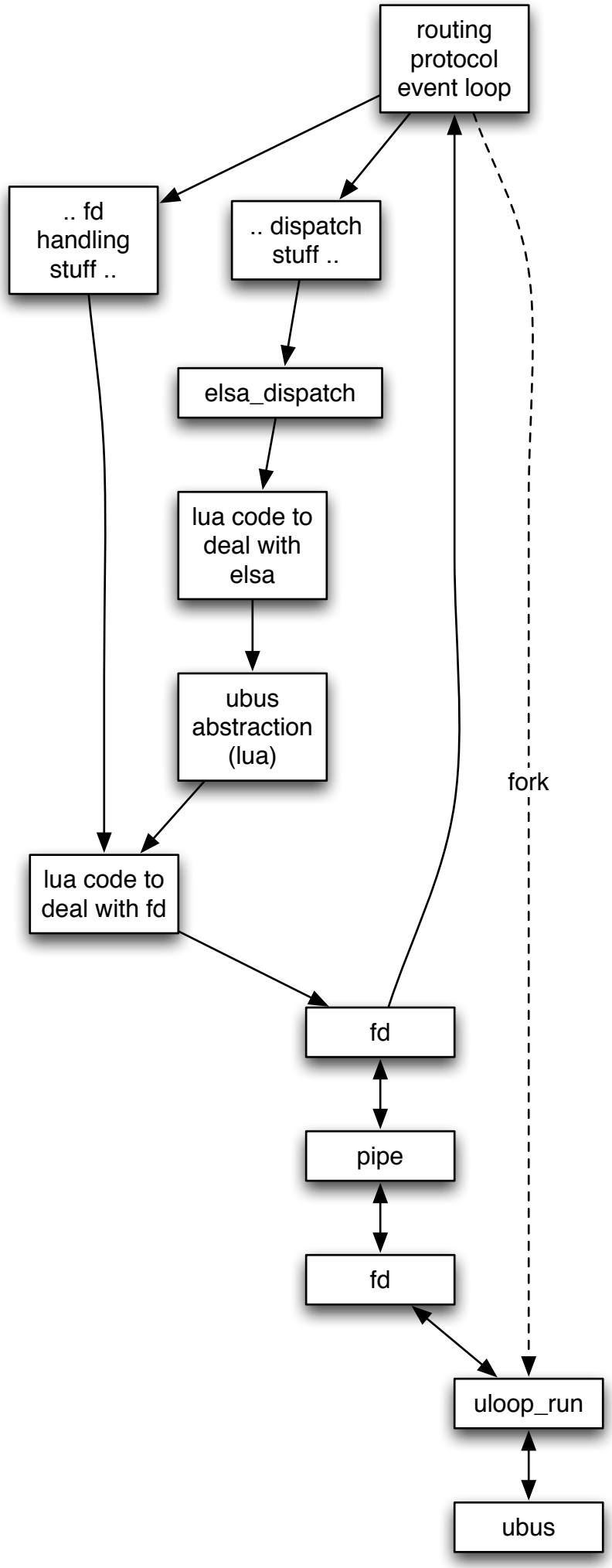
hnet repo hierarchy



skv flow (which keys go where) - read elsa_pa.lua + code under spec/ for description of key contents

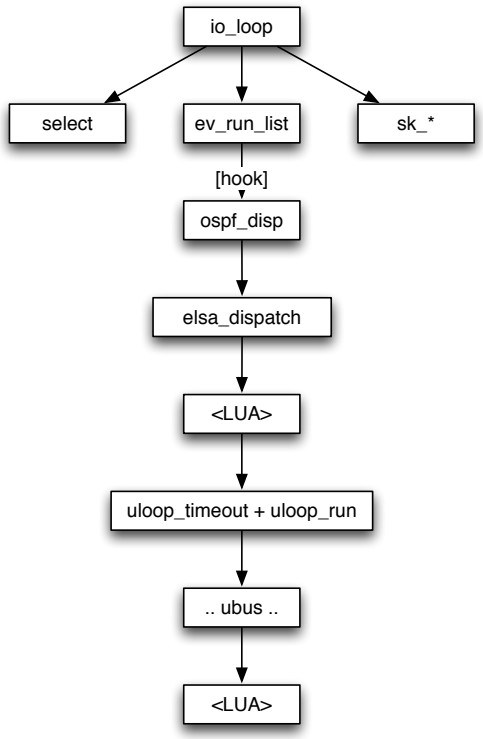


TODO? forked bird idea

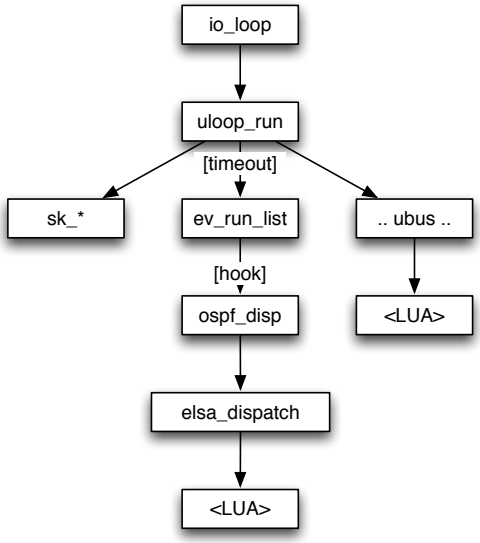
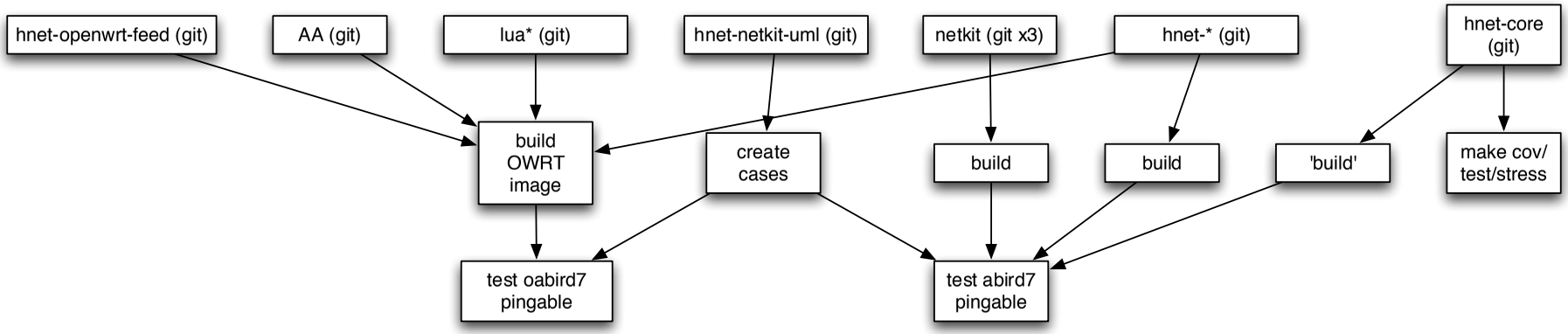


TODO? 201209 bird-elsa-lua-ubus

NOW (Bird+ELSA approach would be ..)



Correct? approach would be to use uloop as base event loop



TODO - continuous integration ideas